Incremental Backedges

- Precompilation adds another twist
- Can’t store backedges, invert and flatten graph, then invert again to restore
- Validate edges:
  - Check that method intersection wouldn't have returned a new method
  - ~~Recursively verify that the target result also wasn’t affected by the addition of a new method~~ (flattened)
  - Unlike typical usage, world counter on reload is not ordered. More expensive and complex world comparison required.
  - Computed by `lowerbound_dependent_world_set`: given an ordered world in the current process, determine the nearest unordered world that was visible in the compile process

Recomputing Validity

- Given a list of invalidated edges
- Decide if the method is still valid
- TODO: implement 😅